

REMARKS

In the Office Action, the Examiner noted that Claims 1 through 16 were pending in the Application. The Examiner rejected all claims. Applicants traverse the rejections below.

I. Traversal of the Rejections over the Cited Art

The Examiner rejected Claims 1, 2, 4 - 8 and 10 - 16 under 35 U.S.C. 102(e) as being anticipated by U.S. Patent No. 6,286,035 to Gillis et al (Gillis). The Examiner rejected Claims 3 and 9 under 35 U.S.C. 103(a) as being unpatentable over Gillis as applied in Claims 2 and 8 in view of U.S. Patent No. 5,361,351 To Lenkov et al (Lenkov). Applicants traverse this rejection below.

A. The Present Invention

The present invention provides a technique for recognizing command related items in a body of object code. A list of required command names and/or option names in programming language form are entered into a filter table. The object code is then scanned for all bit strings potentially representing command names and identifying such names. For each potential command name, succeeding bits are then examined for bits which represent valid options for each of the command name so as to further confirm the identity of commands having valid combinations of command names and/or options. Optionally, the command/option syntax may also be checked. The candidate commands, including options, now identified in programming language form are then compared with the entries in the filter table to determine whether or not they match the required list, which may be further refined by use of logical operators.

B. Differences Between the Present Claims and the Cited Art

US Patent No. 6,268,035 (Gillis) describes parsing and validating command and configuration messages received across a network at run-time. Gillis does not disclose recognizing command related items in a body of object code, and in particular Gillis does not disclose scanning a body of object code for bit strings representing command names.

Claim 1 recites “entering a list of entries, each comprising a required command name and/or option names in programming language textual form, into a filter table”. Relative to this subject matter, Column 2, lines 35-45 is cited. This passage states that “a potentially valid command message” is received “from a network element in response to a command request” and “the potentially valid command message” is stored. “The system parses and validates the command message to determine whether the command message is valid. Parsing and validation take place by accessing tables which contain all relevant information about the allowed parameters for command messages and if the command message is determined to comprise valid parameters, a data structure is populated with the valid parameters.”

This does not disclose entering a list of entries, each comprising a required command name and/or option names in a programming language textual form, into a filter table, as per the first element of Claim 1. Tables are described in Gillis which contain all relevant information about allowed parameters for command messages. There is no discussion about what a command name is relative to a command message of Gillis. In Gillis, the command message is received from a network, and a determination is made as to whether the message is valid. The Gillis ‘command message’ does not come from a body of source code, as will be further discussed below. So it would appear that the Gillis command message is something different.

Claim 1 also recites “scanning the body of object code for all bit strings potentially representing command names and identifying such command names”. Relative to this subject matter, the Office Action cites passages from Column 3, lines 40-50 and Column 4, lines 45-50. The first passage discussed message sets and message names and what they are. The second passage states that a “compiled message parsing engine is stored as object code.” Neither

passage teaches, suggests or discloses that a body of object code is scanned for all bit strings potentially representing command names and that such command names are identified.

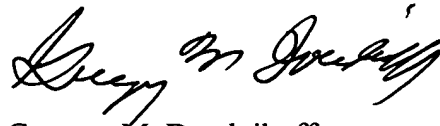
Claim 1 also recites that “for said identified commands, comparing said identified command names and/or option names in programming language textual form with the entries of said filter table to determine whether or not they match any of the list of required command names and/or options in said filter table.” Relative to this subject matter, passages from Column 4, lines 26 - 30 and Column 7, lines 62 - 64 are cited. The first passage discusses tables 70 that are used to perform a message validation procedure. The second passage discusses determining if an associated parameter description is found using a unique key of the received parameter as compared to the data found in the tables. No identified commands are discussed. The programming language textual form is not discussed. Matches to a list of required command names is not discussed.

Accordingly, Applicants submit that independent Claim 1 patentably distinguishes over Gillis. There is simply no scanning of a body of object code described in Gillis. Independent Claims 7 and 14 were rejected for the same reasons as was Claim 1, so it follows that these claims as well as all of the dependent claims patentably distinguish over Gillis.

II. Summary

Applicants have presented technical explanations and arguments fully supporting their position that the pending claims contain subject matter which is not taught, suggested or disclosed by the cited art. Accordingly, Applicants submit that the present Application is in a condition for Allowance. Reconsideration of the claims and a Notice of Allowance are earnestly solicited.

Respectfully submitted,

A handwritten signature in black ink, appearing to read 'Gregory M. Doudnikoff', written in a cursive style.

Gregory M. Doudnikoff
Attorney for Applicant
Reg. No. 32,847

GMD/ssc

Docket No:GB920000081US1

PHONE: 919-254-1288 FAX: 919-254-4330

GB920000081US1